Tecnura
ISSN: 0123-921X
tecnura@udistrital.edu.co
Universidad Distrital Francisco José de Caldas
Colombia

Salcedo Parra, Octavio J.; Hernández, Cesar; Manta C., Hector C.
Analisis y evaluación del routing information protocol RIP
Tecnura, vol. 14, núm. 27, julio-diciembre, 2010, pp. 89-108
Universidad Distrital Francisco José de Caldas
Bogotá, Colombia

Analisis y evaluación del routing information protocol RIP

Analysis and evaluation routing information protocol RIP

Octavio J. Salcedo Parra
Ingeniero de Sistemas, Magíster en Teleinformatica, Magíster en Economía. Docente de la Universidad Distrital Francisco José de Caldas. Bogotá, Colombia.
osalcedo@udistrital.edu.co

Cesar Hernández
Ingeniero Electrónico, Especialista en Servicios Telemáticos e Interconexión de Redes, Magister en Ciencias de la Información y las Comunicaciones. Docente e investigador de la Universidad Distrital Francisco José de Caldas. Bogotá, Colombia. cahernandezs@udistrital.edu.co

Hector C. Manta C.
Ingeniero Electrónico, Magíster en Ciencias de la Información y las Comunicaciones. CCNP e Ingeniero de Telmex. Bogotá, Colombia. hcmantac@udistrital.edu.co

Clasificación del artículo: investigación (recreaciones)

Fecha de recepción: noviembre 29 de 2009

Fecha de aceptación: mayo 25 de 2010

Palabras clave: Enrutamiento, Métrica, Reloj, RIP.

Key words: Routing, Metric, Clock, RIP.

Resumen

Este documento presenta las características del protocolo RIP, versiones 1, 2 y RIPng, y se analiza la configuración y la solución de los problemas e hipótesis sobre este protocolo en algunos de los entornos de red básicos. Se trata de un método de enrutamiento de hace varios años, que se ha convertido en un estándar basado en un algoritmo de vector de distancia destinado a redes redundantes y pequeñas que sigue utilizándose de manera extendida.

Abstract

This paper presents the characteristics of the RIP protocol, versions 1, 2 and RIPng, and discusses the configuration and solution of problems and hypotheses for this protocol in some of the basic network environments. This is a routing method made several years ago, which has become a standard that is still used widely, based on a distance vector algorithm for redundant networks and small.

* * *

1. Introducción

El protocolo de enrutamiento de información se desarrolló en 1970 en los laboratorios de Xerox como parte de otro protocolo de enrutamiento y su popularidad se debe a que fue distribuido con el UNIX de la universidad de Berkeley.

2. Funcionamiento del protocolo

El funcionamiento de RIP es muy fácil y tiene en cuenta ciertas normas elementales:

- Cuando un enrutador se inicializa, las únicas rutas de lasque tiene constancia son las redes a las que está directamente conectado.
- En la versión 1 del protocolo RIP, el enrutador transmiten formación sobre todas las redes directamente conectadas. Estas difusiones se conocen como *triggered updates* (actualizaciones o anuncios).
- Los enrutadores RIP "escuchan" las difusiones RIP; de esta manera pueden informarse de las redes que no tengan constancia directamente.
- La métrica utilizada se basa en el número de saltos (número de enrutadores presentes en una ruta), los cuales se anuncian en cada difusión que se efectúa en cada red.
- Se supone que cualquier ruta que conozca un enrutador RIP pasa por dicho enrutador. Es decir, si el enrutador A envía una actualización al enrutador B, este último supone que el salto siguiente corresponde a las redes que se incluyen en la actualización es el enrutador A.
- Las actualizaciones se envían en intervalos regulares (30 segundos).
- RIP utiliza UDP para enviar sus mensajes y el puerto520.
- La métrica de un destino se calcula como la métrica comunicada por un vecino más la distancia en alcanzar a ese vecino. Esto, teniendo en cuenta el límite de 15 saltos mencionado anteriormente.
- Las métricas se actualizan solo en el caso de que la métrica anunciada más el coste en alcanzar sea estrictamente menor a la almacenada. Solo se actualizará a una métrica mayor si proviene del enrutador que anunció esa ruta.
- Las rutas tienen un tiempo de vida de 180 segundos. Sí pasado este tiempo no se han recibido mensajes que confirmen que esa ruta está activa, se borra. Estos 180 segundos, corresponden a 6 intercambios de información.

2.1. Ejemplo Protocolo RIP

La Figura 1 presenta el diseño para ofrecer un ejemplo del funcionamiento de RIP, en el cual se presentan cuatro enrutadores, con sus respectivas tablas de enrutamiento, las pasarelas iniciales que se deben configurar de manera estática y ocho redes [1].

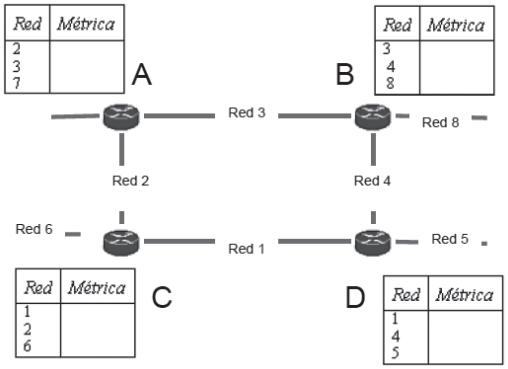

Figura 1. Diseño de red para el funcionamiento del RIP.

Así, el enrutador D se inicializa y envía su primera difusión a las redes adyacentes a él, pero él no conoce si existen o no enrutadores a los cuales este envía la actualización, porque parte del hecho de que hay algún dispositivo escuchando al otro lado. Las redes que sí presentan enrutador reciben la actualización y determinan la ruta más corta, rechazando en este ejemplo la ruta más larga en su tabla. Vemos que en la actualización se envían rutas con una métrica mayor para otros enrutadores (Figura 2), por lo cual no se incluye en la tabla.

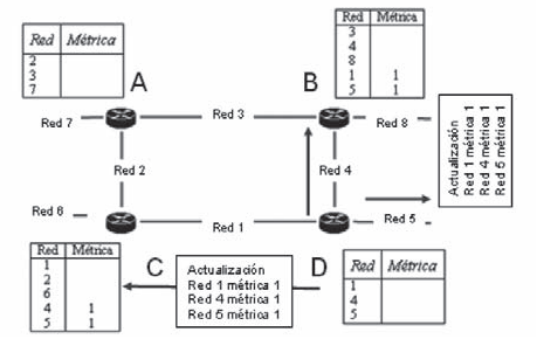

Figura 2. Métricas para las rutas.

C y B ahora transmiten la información sobre todas las redes que conocen a todas las subredes que están ahora conectadas. El enrutador A recibe las actualizaciones y observa que se emiten rutas de igual métrica a la red 5 (Figura 3); así A solo recibe la primera en llegar.

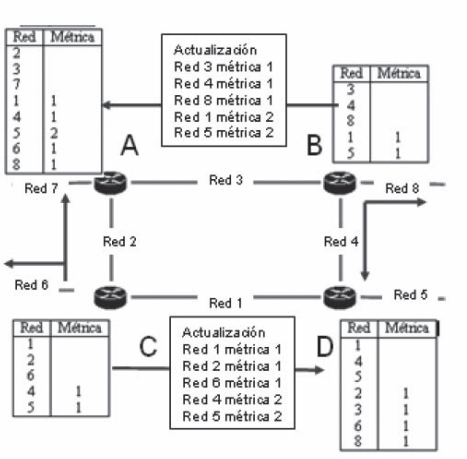

Figura 3. Métricas entre las rutas.

Finalmente A transmite la información sobre sus redes en la actualización siguiente. B y C informan a D sobre estas dos nuevas rutas (Figura 4), lográndose así la convergencia [1].

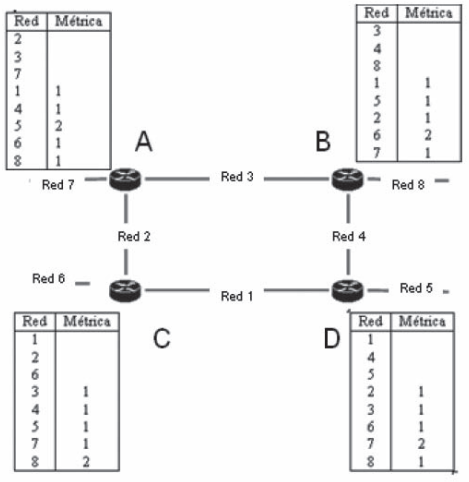

Figura 4. Transmisión de la información sobre las redes.

2.2. Algoritmo vector de distancia

Se basa en el hecho de que "enrutar es la tarea de encontrar una ruta de un remitente a un destino deseado" [1] y lo hace por el intercambio de solo una cantidad pequeña de información, manteniendo una tabla o vector que le indica la distancia mínima conocida hacia cada posible destino y qué línea o interfaz debe utilizar para llegar a él.

Las características que presenta este algoritmo son:

- Datos iniciales en un enrutador: métrica a sus vecinos.
- Lista de parejas (vector o destino, métrica).
- Cada enrutador envía a sus vecinos todas las parejas que conoce, cada cierto tiempo.
- Con esa información, cada enrutador decide el mejor camino a cada destino.
- Problema de convergencia lenta o cuenta hasta infinito (solución: infinito=distancia máxima+1).

2.3. Horizonte partido (splithorizon)

Básicamente consiste en nunca publicar una ruta hacia la interfaz desde la cual se aprendió, ya que así prevendrá cualquier bucle del enrutamiento; esto involucra solo dos vías de acceso.

Es una herramienta implementada en el enrutamiento de vectores de distancia para reducirla aparición de bucles de enrutamiento, garantizando que las rutas comunicadas a través de una interfaz dada nunca se difundan desde la misma interfaz. El *horizonte partido con envenamiento inverso* reduce el tiempo de convergencia, en caso de que se produzca un bucle de enrutamiento mediante el enuncio de una métrica infinita en una interfaz dada para las rutas difundidas a través de dicha interfaz.

2.4. Relojes de actualización

Sirven para que el enrutador sepa cuándo debe esperar antes de enviar las actualizaciones periódicas. En la versión 1 del protocolo RIP, cada actualización incluye todas las rutas (salvo las que haya eliminado el horizonte partido), independientemente de si se han producido cambios desde la última actualización. Este procedimiento periódico de actualizaciones garantiza que los enrutadores puedan determinar si otros enrutadores están apagados. Sin embargo, el breve periodo de tiempo que el protocolo RIP espera entre dos actualizaciones, junto con el hecho de que de cada actualización se anuncia toda la tabla de enrutamiento, indica que el protocolo puede utilizar buena parte de ancho de banda en redes complejas.

2.5. Relojes de espera

Los relojes de espera también sirven para evitarlos bucles en una topología compleja, al solicitar que un enrutador RIP espere un periodo de tiempo específico (por omisión 180 segundos)antes de considerar verdadera cualquier información sobre una ruta actualizada.

2.6. Relojes de eliminación de ruta

El reloj de tiempo inválido de ruta (denominado reloj inválido en IOS) se usa para determinar cuándo ha fallado una ruta. Si una actualización no ha oído acerca de una ruta dada antes de que este reloj caduque, dicha ruta se considera no válida y entra en fase de espera. Sin embargo, sigue siendo utilizada (pero ya no se anuncia) hasta que caduca el reloj de eliminación de ruta; así, cuando caduca este reloj se elimina por completo la ruta de la tabla de enrutamiento.

2.7. Actualizaciones provocadas

Estas sirven para reducir la posible aparición de bucles de enrutamiento y el tiempo de convergencia de la red. Si falla un enlace en una red directamente conectada, en lugar de esperar que caduque un reloj de actualización, el protocolo RIP anuncia un fallo inmediatamente (con una distancia infinita). De esta manera, una vez que se ha actualizado una ruta, el protocolo RIP anuncia a continuación la ruta actualizada, en lugar de esperar a que caduque el tiempo de actualización; las actualizaciones provocadas se usan conjuntamente con el envenenamiento inverso de la ruta para propagar rápidamente el fallo de una ruta en cuestión.

2.8. Tipo de encapsulamiento

El encapsulado depende de la interfaz, sea Ethernet, FrameRelay, ATMu otro.

La configuración de encapsulado Ethernet (Fast Ethernet, Gigabit Ethernet) es relativamente sencilla, ya que suele no ser necesario establecerlo si se usa Ethernet estándar (DIX) [2]. El encapsulado por omisión para las interfaces Ethernet (IP) en los enrutadores de CISCO es el encapsulado DIX(conocido como ARPA). Para la interfaces IPX, el encapsulado por omisión es 802.3 (en cisco conocido como Novell-Ether).

Para la configuración FrameRelay, que suele ser también bastante sencilla, es preciso saber qué tipo de encapsulado está utilizando el proveedor: Cisco o IETF. Una vez se dispone de esta información, basta con introducir en modo de configuración de interfaz para la interfaz serie y emitir el comando *encapsulationframe-relay[ietf]*. Si se está utilizando el encapsulado de Cisco, solo hay que introducir *encapsulationframe-relay.*

2.9. Problemas del protocolo

2.9.1. Cuenta infinita

Para evidenciar este problema lo más apto es mostrar en un escenario cómo se presenta. Partamos del hecho de que no se utiliza el horizonte divido o relojes de espera en la malla de la Figura 5, a la cual le corresponden las Tablas 1 y 2 de enrutamiento.

Tabla 1. Tablas de enrutamiento.

	Red	Salto siguiente	Métrica
A	10.0.0.0 172.16.0.0	–	–
B	10.0.0.0 192.168.1.0	–	–
C	192.168.1.0 172.16.0.0 192.168.50.0	–	–

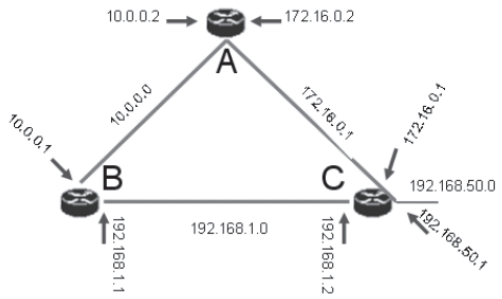

Figura 5. Transmisión de la información sobre las redes.

Tabla 2. Tablas de enrutamiento.

	Red	Salto siguiente	Métrica
A	10.0.0.0 172.16.0.0 192.168.1.0 192.168.50.0	_ _ 10.0.0.1 172.16.0.1	_ _ 1 1
B	10.0.0.0 172.16.0.0 192.168.1.0 192.168.50.0	_ 10.0.0.2 _ 192.168.1.2	_ 1 _ 1
C	10.0.0.0 192.168.1.0 172.16.0.0 192.168.50.0	172.16.0.2 _ _ _	1 _ _ _

Ahora supongamos que falla el enlace de C a la red 192.168.50.0; por ser un enlace directo de C se eliminaría inmediatamente, ya que cada enrutador tiene certeza de sus enlaces pero confía en la veracidad de los demás. Así, este no informaría a A y B de esta eliminación y se tendría que esperar a que caduque el tiempo muerto correspondiente a la red del fallo (180 segundos), para que se deje de anunciar esta red. Sin embargo, A y B siguen actualizando los enlaces de C con información de ruta hacia la red 192.168.50.0, de la cual supusimos el daño; con ello se presenta una adición de ruta equivocada en la tabla de enrutamiento de C (Tabla 3).

Tabla 3. Tablas de enrutamiento de C.

	Red	Salto siguiente	Métrica
C	10.0.0.0 192.168.1.0 172.16.0.0 192.168.50.0	172.16.0.2 _ _ 192.168.1.1	1 _ _ 2

Utilizando uno de los demás enrutadores como salto siguiente, los enrutadores B y C se envían paquetes entre sí hasta que caduca el TTL y, tres minutos más tarde, el enrutador B reconocerá que la ruta que pasa por C ya no es válida, eliminándola de su tabla. A presentaría el mismo problema por lo que procede de la misma manera que B.

Pero el problema persiste, ya que C sigue anunciando la red 192.168.50.0 con una métrica de 3, hacia A y B, y esto a su vez hacia C, lo cual lleva a caducar la ruta de métrica de 3.Por su parte, C se actualiza con métrica de4 con destino a la red del problema; así sucede este problema hasta que la métrica de la ruta ascienda a 16 (aproximadamente una hora más tarde). Cuando llega a 16 es precisamente el momento en el que se presenta el problema de cuanta al infinito que es el inconveniente principal que plantea el enrutamiento de vectores de distancia [3].

2.9.2. Solución cuenta infinita

Las soluciones ya han sido expuestas, pero no en aplicadas a la práctica. Empecemos con la activación del horizonte partido; si se hubiera activado en C, en el momento de fallo de la red 192.168.50.0 no hubiera recibido actualización de la ruta falsa de A o B, evitando actualizaciones de esta ruta sobre las redes 192.168.1.0 y 172.16.0.0.

Pero entre A y B persiste el problema de conteo al infinito, anunciándose mutuamente la ruta 192.168.50.0 a través del enlace 10.0.0.0.Así, añadir relojes de espera resolvería este problema. De esta forma, cuando caduque el tiempo muerto de B con respecto a la red192.168.50.0, B podrá poner la ruta en tiempo muerto en la que, durante 180 segundos, se ignora cualquier actualización relacionada con la red 192.168.50.0. En el momento que se elimina la ruta de tiempo muerto, el enrutador A se percata del fallo de la ruta y evita el bucle.

Con las actualizaciones provocadas y el envenenamiento de ruta, tan pronto cuando falle el enlace de C, este anunciará de inmediato (en la actualización provocada) una métrica infinita (en un envenenamiento de ruta) para la red 192.168.50.0, dirigida a los enrutadores A y B. Estos hacen caducar la ruta e introducen en tiempo muerto.

2.10. Especificaciones del protocolo

Se supone que cada computadora central que lleva a cabo los RIP tiene una tabla de enrutamiento. Esta tabla tiene una entrada para cada destino alcanzable a través del sistema descrito por los RIP. Cada entrada contiene por lo menos la siguiente información:

- La dirección IP del destino.
- Una métrica que representa el costo total de conseguir que un datagrama de la computadora central llegue al destino. Este métrico es la suma de los costos asociados con las redes que cruzaría consiguiendo el destino.
- La dirección IP de la próxima vía de acceso a lo largo de la ruta al destino. Si el destino es sobre las redes directamente conectadas, este elemento no se necesita.
- Un señalador para indicar información sobre la ruta cambiada recientemente. Esto es llamado el señalador de cambio de ruta.
- Los relojes se asociaron con la ruta.

2.11. Formato del mensaje

Cada datagrama (Figura 6) contiene un comando, un número de versión y los posibles argumentos.

En el campo de comando se puede dar:

- Solicitud**:** El sistema responde para enviar a todos, parte de su tabla del enrutamiento.
- Respuesta**:** Un mensaje que contiene todo o parte del remitente de la tabla de enrutando. Este mensaje puede enviarse en respuesta a una solicitud, o puede ser un mensaje de la actualización generado por el remitente.
- Traceonobsoleto: Mensajes que contienen este comando son para ignorarlos.
- Traceoffobsoleto**:** Mensajes que contienen este comando pueden ser ignorados.
- Reservado: Este valor se usa por Sun Microsystems para sus propios propósitos. En el campo de versión se presenta el proceso de entrada.
- Proceso de entrada: Presta un manejo de datagramas recibidos en UDP, y se determina según su valor 0, 1 o mayor que 1:
- 0 datagramas cuyo número de versión es el cero, será ignorado.
- 1 datagrama cuyo número de versión es uno, será procesado.

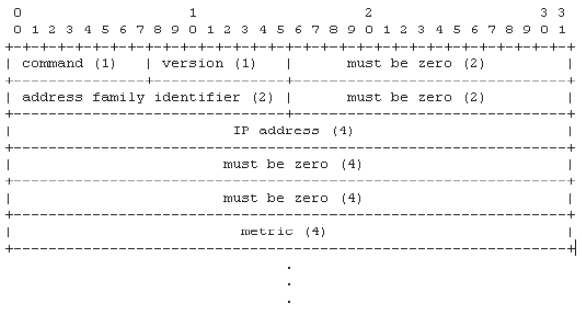

Figura 6. Formato del mensaje RIP.

- >1 datagramas cuyo número de versión es mayor a uno, es procesado. Las versiones futuras del protocolo pueden poner los datos en estos campos.

3. RIP versión 2

Lo que hasta ahora se ha descrito se basa en el estándar RIP [2]. En 1994se presenta una serie de extensiones que mejoran las falencias de RIP en aspectos como autenticación y soporte de Máscara de Subred de Longitud Variable (VLSM).

3.1. Formato del mensajeRIPv2

Los primeros cuatro octetos de un mensaje del RIP contienen el encabezado del RIP. El resto del mensaje está compuesto de 1 a 25 entradas de la ruta (20los octetos cada uno). El nuevo formato de mensaje de RIP se muestra en la Figura 7:

Los campos comando, dirección del identificador familiar (AFI), dirección IP, y métrico están definidos en RFC 1058 [2].

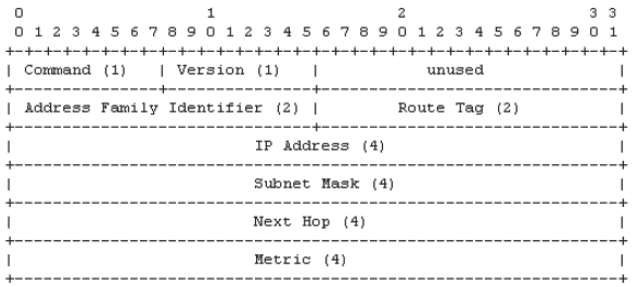

Figura 7. Nuevo formato del mensaje RIP.

3.2. Autenticación

El esquema de la autenticación requerirá más de dos octetos. La autenticación para los RIP versión 2 usará el espacio de la entrada del RIP entera (Figura 8). Si el identificador de dirección familiar (*addressfamilyidentifier*) que está de primero (y solo el primero) en la entrada en el mensaje es el *0xFFFF*, entonces el resto de la entrada contiene la autenticación.

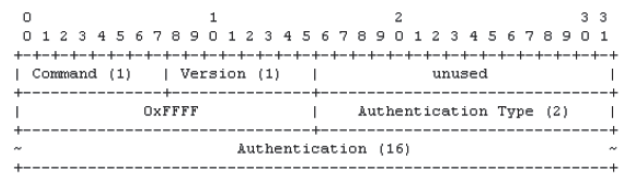

Figura 8. Esquema de autenticación.

El único tipo de la autenticación es la contraseña simple que permanece en 16 octetos que contienen la contraseña del texto plano. Si la contraseña está bajo 16 octetos, debe justificarse a la izquierda y debe llenarse a la derecha con valores *nulls* (el 0x00).

De esta manera se admite la autenticación de texto no cifrado para los enrutadores compatibles con RFC(los enrutadores de CISCO también admiten la autenticación cifrada MD5).

3.3. Etiqueta de ruta

La etiqueta de la ruta (RT) del campo es un atributo asignado a una ruta que debe preservarse. El uso intencional de la etiqueta de la ruta es proporcionar un método de separar los RIP "interiores" de las rutas (las rutas para las redes dentro del dominio de enrutamiento de RIP) de los RIPS "externos" que se pueden haber importado de un EGP u otro IGP. Esto permite la posibilidad de un BGP-RIP, es decir, las interacciones protocolares que describiría los métodos para sincronizar el enrutamiento en una red.

3.4. Máscara de subred

Las máscaras de subred se transmiten con las actualizaciones de la versión 2 del protocolo. Si el campo (*subsetmask*) es el cero, entonces ninguna máscara de la subred ha sido incluida para esta entrada. Para ello se aplican las siguientes reglas:

1) La información interior a una red no debe anunciarse en otra red.

2) La información sobre una subred no puede anunciarse donde los enrutadores RIP-1 la considerarían una ruta de la computadora central.

3) Las rutas de súperredes (las rutas con un *netmask* menos específica que la máscara de la red "natural") no debe anunciarse donde ellas pudieran ser malinterpretadas por los enrutadores RIP-1.

3.5. Próximo salto

Es una dirección específica como un próximo salto que debe, por la fuerza, alcanzar directamente la subred por encima del anuncio hecho.

El propósito del próximo campo del salto es eliminar el paquete enrutando, a través de los saltos extras en el sistema. Es particularmente útil cuando el RIP no está ejecutándose en todos los enrutadores en una red.

3.6. Actualizaciones multidifundidas

Las actualizaciones retransmiten mediante multidifusión, en lugar de utilizar la difusión normal, ahorrando así el número de ciclos de UPC a los servidores no RIP.

4. RIP para IP v6

RIPng es un protocolo basado UDP. Cada enrutador que usa RIPng tiene un proceso del enrutamiento que envía y recibe datagramas en el puerto 521.

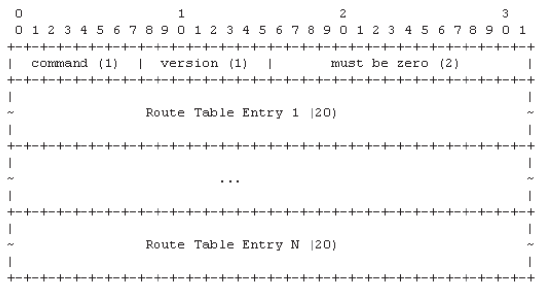

Figura 9. Formato RIP sobre IPV6.

Como se observa en la Figura 9, este formato es similar al formato de las versiones anteriores, pero para cada uno de los tipos del mensaje, el resto del datagrama contiene una lista de RTE (tabla de la ruta entrada). Cada RTE en esta lista contiene un prefijo del destino, el número de bites significantes en el prefijo, y el costo para alcanzar ese destino (métrico), donde cada entrada de tabla de ruta (RTE) tiene el siguiente formato (Figura 10):

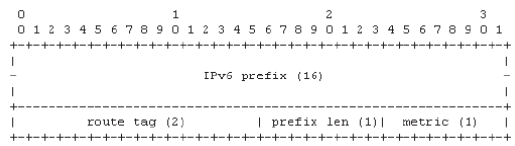

Figura 10. Formato de la entrada de tabla de ruta.

También presenta un próximo salto de ruta con el siguiente formato (Figura 11):

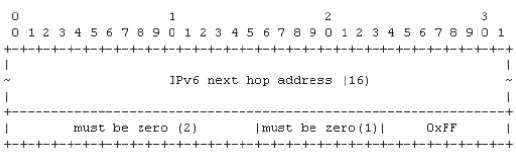

Figura 11. Formato del próximo salto de ruta.

Especificando un valor de 0:0:0:0:0:0:0:0 en el campo del prefijo de un próximo salto, RTE indica que la próxima dirección de salto debe ser la originadora del anuncio de RIPng. Una dirección especificada como un próximo salto debe ser una dirección local de enlace.

El propósito del próximo salto RTE es eliminar paquetes que están siendo ruteados a través de los saltos extras en el sistema. Esto es particularmente útil cuando RIPng no está ejecutándose en todos los enrutadores en una red. Los formatos de paquete de RIP (versión 1 y 2) no distinguen entre los varios tipos de la dirección; la distinción entre la red, subred y las rutas de las terminales no necesitan ser hechas para RIPng porque en una direcciónIPv6 el prefijo es inequívoco[6].

5. Configuración de RIP

Para configurarlo, utilizando las opciones por omisión, basta con emplear dos comandos *routerripynetwork [dirección de red*]. Cuando se introduce el comando *routerrip* desde el modo de configuración global [1,4], se activa el protocolo RIP de forma global y se sitúa en el modo de configuración de enrutador; mostrado así:

(config)#router rip

(config-router)#

Luego se introduce el comando *network* para activar de forma individual el enrutamiento RIP para cada red basada en clases. En este comando:

1. Se anuncian las rutas pertenecientes a la red específica basada en clases.

2. Se escuchan todas las actualizaciones en todas las interfaces pertenecientes a la red basada en las clases en cuestión.

3. Se envían actualizaciones en todas las interfaces pertenecientes a la red basada en las clases en cuestión.

Si se desea modificar la versión RIP que se está utilizando, se usa el comando, *versión [1 / 2]* oi*prip[sendversion / receiveversion] [1 / 2*]

Como se puede observar, conseguir que el protocolo RIP funcione utilizando las opciones por omisión no es en absoluto difícil. Sin embargo, si se desea optimizar el manejo del protocolo hay que seguir unos cuantos pasos adicionales para configurar diversas tareas opcionales como [5]:

- Configuración de interfaces pasivas
- Configuración de actualizaciones unidifusión
- Incorporación de compensación métrica en las rutas
- Ajuste de relojes RIP
- Desactivación de horizonte partido
- Establecimiento de número máximo de rutas
- Configuración de autenticación (RIP 2)
- Desactivación de auto resumen (RIP 2)

5.1. Ejemplo de configuración

Se presenta un ejemplo típico de configuración de RIP con tres enrutadores con sus respectivas interfaces activadas. Se hace de manera adicional un pedido de encapsulación sobre FrameRelay y una no configuración del horizonte partido sobre los enrutadores A y B [4]. Este escenario aparece en la Figura 12:

```
Configuración para Router A
interfaceethernet 1
ip address 12.13.50.1
!
interface serial 1
ip address 128.125.1.2
encapsulation frame-relay
noip split-horizon

Configuración para Router B
interfaceethernet 2
ip address 20.155.120.1
!
interface serial 2
ip address 131.108.1.2
encapsulation frame-relay
noipsplit-horizon

Configuración para Router C
interfaceethernet 0
ip address 10.20.40.1
!
interface serial 0
ip address 128.124.1.1
ip address 131.108.1.1 secondary
encapsulation frame-relay
```

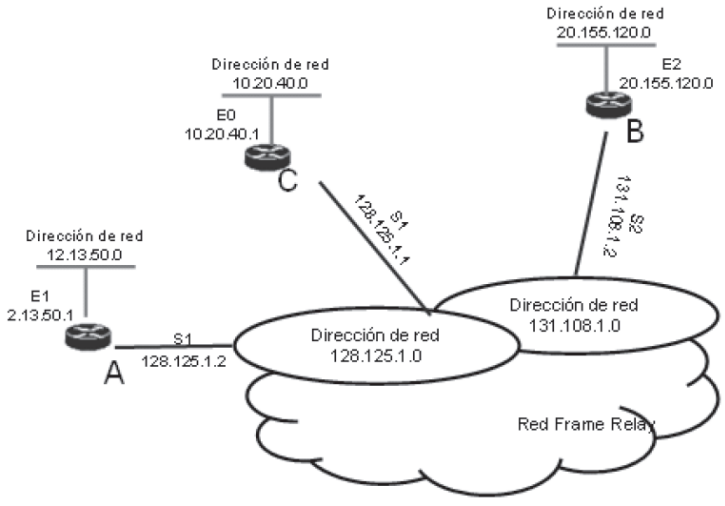

Figura 12. Escenario de configuración de RIP.

6. Configuración avanzada y optimización del protocolo RIP

Las siguientes son opciones de configuración que resultarán muy útiles en ciertos casos; estas opciones de configuración son:

- Configuración de interfaces pasivas
- Configuración de actualizaciones unidifusión
- Incorporación de compensaciones métricas a las rutas
- Ajustes de los relojes RIP
- Desactivación del horizonte partido
- Establecimiento del número máximo de rutas
- Configuración de autenticaciones(RIPv2)
- Desactivación de auto resumen

6.1. Configuración de interfaces pasivas

Una interfaz pasiva es una interfaz que no difunde actualizaciones de enrutamiento, pero que sigue anunciándose en dichas actualizaciones y también sigue escuchando cuando estas aparecen. El comando *passive interface* suspende la tarea de las difusiones del comando *network* sobre una interfaz específica. Esto es útil cuando:

La red vinculada a la interfaz RIP solo incluye servidores pero debe ser anunciada a otros enrutadores Por razones de seguridad o rendimiento conviene desactivarlas actualizaciones de enrutamiento difundidas y elegir selectivamente los enrutadores que recibirán actualizaciones unidifusión

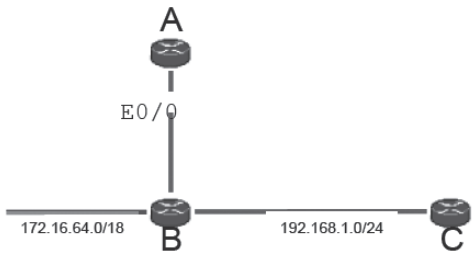

Figura 13. Esquema de una red.

En la Figura13 la red 172.16.64.0/18 solo contiene servidores. Las difusiones RIP en esta red estarían demás, pero si no se introduce un comando de red para la red 172.16.0.0, no solo no se anunciará la red 172.16.64.0/24 al enrutador A o C, sino que la red172.16.128.0/18 tampoco se anunciará. Además, el enrutador B no anunciará ninguna red a A, lo que significa que el enrutador A nunca sabrá de la existencia de la red192.168.1.0/24 ni de la red172.16.128.0/18, a la vez que el enrutador C tampoco sabrá de la existencia de la red 172.16.128.0/18.

Para resolver este problema hay que usar el comando de interfaz pasiva así: *passive interface [tipo y numero de interfaz]*. Se introduce el comando desde el modo de configuración de enrutador de la siguiente manera:

```
B(config)# router rip
B(config-router)# network 172.16.0.0
B(config-router)#network 192.168.1.0
B(config-router)#passive-interface Ethernet 0/0
```

Después de seguir estos pasos de configuración, las funciones del protocolo RIP del enrutador C serán:

- Se anunciarán todas las redes al enrutadores A y C
- En todas las interfaces se escucharán las actualizaciones de enrutamiento

- Las actualizaciones de enrutamiento no se anuncian en la interfaz E0/0 del enrutador B

6.2. Configuración de actualizaciones unidifusión

Aunque el protocolo RIPv1 suele ser unidifundido para las actualizaciones de enrutamiento y el protocolo RIPv2 usa en su lugar multidifusiones, en ciertas situaciones resulta necesario activar actualizaciones unidifusión. Es en estos casos cuando se envían actualizaciones a través de un enlace que no admite difusiones (redes NBMA-framerelay). También resultan útiles cuando no se desea que las difusiones derrochen recursos de UPC en clientes vinculados a la misma red, como el enrutador que requiere el envío de difusiones. Finalmente, las actualizaciones unidifusión son útiles en situaciones en las que se busca seguridad entre varios enrutadores para las respectivas actualizaciones. Como las actualizaciones para cada enrutador son unidifusión, en una red conmutada, los servidores normales no podrían utilizar un rastreador para leer los detalles de cada actualización RIP. Para comprender el alcance de la utilidad de las actualizaciones se presenta la Figura 14.

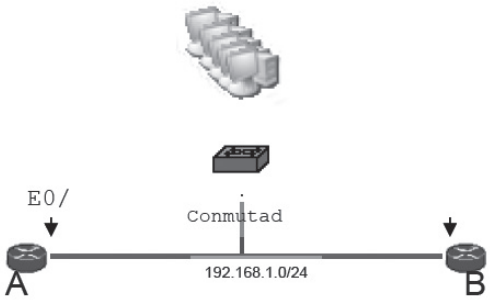

Figura 14. Esquema de actualizaciones.

Así se estuvieran difundiendo actualizaciones, todos los servidores de la red 192.168.1.0 recibirán dichas actualizaciones. En su lugar, basta con introducir solo la dirección IP del *router* A o B y luego pacificar la interfaz E0/0 en ambos enrutadores; es una interfaz pasiva. Esto garantiza que todas las actualizaciones que se produzcan entre A y B sean actualizaciones unidifusión, sin tráfico de difusión RIP presente en la red.

Para activar las actualizaciones hay que usar el comando, del mismo modo que la configuración de enrutador *neighbor [dirección ip]*, por ejemplo:

```
A(config)# router rip
A(config-router)# network 192.168.1.0
A(config-router)# network 192.168.1.1
A(config-router)# passive-interface Ethernet 0/0
```

6.3. Incorporación de compensaciones métricas a las rutas

Añadir una compensación métrica a una ruta (Figura 15) permite especificar que la métrica asignada a las rutas procedentes de un enrutador o red dados aumente una cantidad específica. Esta funcionalidad permite especificar de un modo rudimentario que las rutas procedentes de uno o más enrutadores sean menos favorecidos que otras. [8]

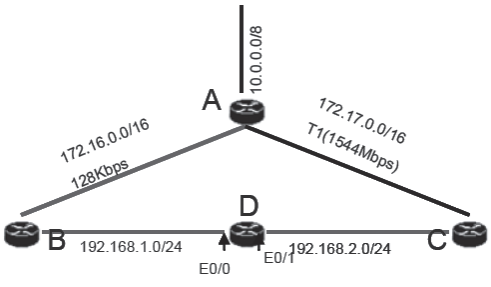

Figura 15. Compensación métrica de rutas.

El inconveniente que presenta esta situación es que el protocolo RIP recibe la misma métrica (dos saltos) para ambas rutas y, por tanto, aplica un

balance de carga a través de las dos, provocando un desbordamiento del *buffer* en el enrutador B y un cuello de botella indirecto para resolver este problema. Cabría especificar que todas las rutas que entren en la interfazE0/0 del enrutador D reciben una compensación métrica de +1; de esta manera, la ruta procedente del enrutador B costaría aparentemente menos que una procedente del enrutador C, y el enrutador D, por su parte, usaría una ruta que pasa por C en lugar de efectuar una balance de carga.

Para añadir una compensación métrica se usa el comando *offset-list[(opcional) lista de acceso] [in | out] [offset] [(opcional) tipo y numero de interfaz]*

6.4. Ajustes de los relojes RIP

Ajustar los relojes de RIP resulta muy útil si se desea optimizar la convergencia de la red; por ejemplo, en una red interna de alta velocidad y ancho de banda considerables (como una LAN de Fast Ethernet), tal vez se desee reducir el valor del os relojes para reducir el uso de ancho de banda, sacrificando el tiempo de convergencia. De una u otra manera, cuando se modifican los relojes, no hay que olvidar que se deben configurar todos los relojes para que usen los mismos valores de reloj, así como deben recordarse las relaciones que establecen los relojes entre sí (Tabla 4).

Tabla 4. Múltiplos de reloj recomendados.

Reloj	Múltiplo	Tiempo por omisión (IP/RIP)
De actualización	Reloj base	30 segundo
No válido	3vecesel de actualización	180 segundos
De espera	3 vecesel deactualización	180 segundos
Eliminación de ruta	Mayor que l no valido	240 segundos

Para establecer los valores de los relojes del protocolo RIP hay que usar el comando de modo de configuración: *enrutador Times Basic [tiempo de actualización en segundos] [tiempo de invalidez en segundos] [tiempo de espera en segundos] [tiempo de eliminación de ruta en segundos].*

Por ejemplo, para asignar al reloj de actualización un valor de 1 segundo, al reloj no válido 45 segundos, al reloj de espera 55 y al reloj de eliminación de ruta 100, se tendría que ejecutar el comando siguiente:

Router (config-router)#timers basic 15 45 55 100

Esta configuración hace que el enrutador envíe y espere recibir actualizaciones cada 15segundos; que declare una ruta inadecuada tras 45 segundos sin actualización y entre en fase de espera; que permanezca en dicha fase durante unos 55 segundos adicionales y luego, 100 segundos más tarde, proceda a eliminarla ruta de la tabla[9].

6.5. Desactivación del horizonte partido

Este procedimiento *no* se recomienda en la mayoría de los casos, debido a que la presencia de este ahorra molestias al evitar bucles de enrutamiento. Sin embargo, si se tienen enlaces WAN con múltiples circuitos virtuales (VC) en una única interfaz física, el horizonte partido puede ser una mala elección.

Hay que suponer que se dispone de una sola interfaz, una serie con cuatro VC FrameRelay y cuatro redes aplicadas a ella. Con el horizonte partido activado no se enviarán actualizaciones recibidas por ninguno de los VC presentes en los enlaces a ninguno de los VC restantes. Este problema tiene fácil solución.

Se deben examinar primero las configuraciones por omisión del IOS para el horizonte partido sobre FrameRelay. Estas configuraciones son:

- El horizonte partido se desactiva si no se definen subinterfaces
- El horizonte partido se activa para las subinterfaces punto a punto
- El horizonte partido se desactiva para las subinterfaces multipunto

El horizonte partido está activado en las conexiones punto a punto[1]; en otras palabras, si se tiene conocimiento de una ruta S0/0.1, esta no se retransmitirá a S0/0.1, pero sí a S0/0/.2.

En el caso de las interfaces sin subinterfaces, si solo se asigna un VC a la interfaz, lo mejores activar el horizonte partido a las interfaces. Si hay asignadas múltiples VC a la interfaz, existen dos opciones:

- Activar el horizonte partido y configurar estáticamente cualquier ruta que se haya propagado
- Dejar desactivado el horizonte partido y filtrar rutas que podrían producir bucles mediante el uso de listas de acceso
- Reconfigurar el enrutador para que utilice subinterfaces

Las primeras dos opciones son problemáticas y no recomendadas; la tercera es la opción preferible en casi todas las ocasiones. En esta simplemente se reconfigura el enrutador para usar subinterfaces para cada VC.

De cualquier manera, activar y desactivar el horizonte partido es tan fácil como introducir el comando *ipsplit-horizon* para activar el horizonte partido, o *no ipsplit-horizon*en el modo de configuración de interfaces para cada interfaz en particular.

[1] En este caso, el protocolo RIP trata cada subinterfaz como una subinterfaz independiente para aplicar el horizonte partido.

6.6. Establecimiento del número máximo de rutas

En el IOS de cisco, la configuración por omisión para todos los protocolos de enrutamiento, salvo BGP[2], es el balance de cargas efectuado, de hasta cuatro rutas de coste equivalente. En algunos casos esta configuración puede provocar un cuello de botella. Para ajustar esta cifra de modo que admita más o menos rutas, se usa el comando de configuración de enrutador *maximum-paths* [número de rutas] para cada protocolo de enrutamiento que se utilice en el enrutador. Se pueden incluir hasta seis rutas para el IP, aunque hay que recordar que él mismo es, por supuesto, uno.

6.7. Configuración de autenticaciones (RIPv2)

En el protocolo RIPv2 se puede garantizar que, para procesar una actualización desde un enrutador vecino, este está autenticado. Esta autenticación ayuda a garantizar que las actualizaciones de enrutamiento solo sean procesadas si proceden de enrutadores "fiables". Para que la autenticación funcione, todos los enrutadores deben usarla misma contraseña. Activar la autenticación en el protocolo RIPv2implicaseguirlos dos pasos siguientes:

- Configurar una cadena clave
- Autorizarla autentificación de texto sin cifrar o MD5

El primer paso en este proceso consiste en configurar una cadena clave[3];para esto, hay que introducir primero el comando *keyChain [nombredecadena]* en

[2] En BGP, el balance de cargas está desactivado por omisión.

[3] Se trata de una lista de "claves" de autenticación o contraseñas que pueden usarse para autenticar el protocolo de enrutamiento.

modo de configuración global, que ejecuta el modo de configuración de cadena clave para cada cadena en cuestión, como se observa a continuación:

3620B (config)#key Chaín test
3620B (config-keychai)#

Posteriormente se usa el comando *key [numero clave]* para empezar a configurar una clave especifica[4]. Luego se usa el comando *key-string* [contraseña] para establecer la contraseña.

Aceptando modificaciones, como querer determinar los tiempos en los que el enrutador debe aceptar la clave, se usa el comando *accept-lifetime* [tiempo inicial[5]] [tiempo final[6]].

Una vez configurados estos parámetros, la clave está lista para poder aplicarse a la autenticación en el protocolo RIPv2.En primer lugar hay que aplicar la cadena clave a una interfaz específica, usando el comando de modo de configuración de interfaz *ipripauthenticationkey-chain [nombre de cadena clave]*. Posteriormente se establece el modo de autenticación para la interfaz que use el comando de modo de configuración de interfaz *ipripauthenticationmode [md5|text]*.[10].

6.8. Desactivación de auto resumen

En RIPv2, las rutas se resumen automáticamente en todas las direcciones de red, siempre que se cumpla con los requisitos de auto resumen; en estos casos, si se anuncia una ruta en una interfaz que tenga una dirección de red basada en clases distintas de la de la ruta que se va anunciar, todas las subredes de la red anunciada se tomarán como una sola entrada para la totalidad de la red basada en clases.

En otras palabras, si se envía una actualización que contenga las redes 172.16.64.0/18 y 172.16.128.0/18 en una interfaz con una dirección IP de 172.31.1.1/20, el anuncio se dirigirá a 172..16.0.0/16 y no a cada subred individual. Esta característica puede causar problemas si se tiene una topología basada en VLSM, pero, si no es el caso, reduce significativamente el número de rutas necesarias que hay que anunciar y mantener en la tabla de enrutamiento.

7. Recomendaciones de uso RIP

Si surgen problemas a la hora de usar el protocolo RIP, conviene tener en cuenta lo siguiente:

- El protocolo RIPv1no admite VLSM. Hay que recordar que, dependiendo de la configuración, el protocoloRIPv1 resume las redes VLSM en una sola dirección de reden clases, o bien se niega por completo a anunciar la ruta.
- La opción de auto resumen está desactivada por omisión en el protocolo RIPv2. Si se está usando dicho protocolo, hay que asegurarse de que la topología de la red puede auto resumirse o, de lo contrario, conviene desactivar el auto resumen.
- Si se activa la autenticación, todos los enrutadores que participen en la red RIP deben usar la misma contraseña.
- Es posible que los enrutadores que no pertenezcan a CISCO no soporten MD5.
- El horizonte partido puede ocasionar problemas en los PVC de FrameRelay, si múltiples PVC aparecen asignados a la misma interfaz. Para eliminar este problema hay que usar subinterfaces con único PVC por cada una.

---

[4] Dentro de un rango [1-4000000000].

[5] Su formato es hh:mm:ss día, mes, año y se especifica el momento en el cual se aceptara la contraseña.

[6] Infinitive para siempre ó duration seguida de un numero de segundos, periodo en segundos después de tiempo inicial, o con el formato hh:mm:ss

- En el caso de los PVC múltiples debe desactivarse el horizonte partido. En topologías multipunto es necesario incluir el filtrado de rutas para prevenir bucles de enrutamiento.

- Hay que garantizar que concuerden todos los relojes presentes en todos los enrutadores de la topología. En caso contrario, las actualizaciones de enrutamiento pueden fallar, formándose a su vez bucles de enrutamiento.

- Hay que asegurarse de que los valores de reloj estén relacionados entre sí.

- No debe olvidarse que el protocolo RIP es proclive a formar cuellos de botella cuando se realiza balance de carga.

- Si no se envían actualizaciones desde una interfaz, hay que asegurarse de haber introducido correctamente las direcciones IP de los enrutadores remotos.

- Hay que comprobar si el problema se debe a una cuestión relacionada con el protocolo RIP y no con un problema de configuración físico o lógico.

8. Ventajas y desventajas de RIP v1

La Tabla 5 presenta un resumen de las ventajas y desventajas de RIP v1.

9. Comparación de RIP con OSPF

Comparar RIP y OSPF no resulta muy adecuado, ya que ambos protocolos se han diseñado para entornos totalmente distintos. OSPF está diseñado para

Tabla 5. Ventajas y desventajas de RIP v1.

Ventajas	Desventajas
Es muy fácil de entender y configurar	Es ineficaz (ocupa demasiado ancho de banda)
Está admitido casi con seguridad en todos los enrutadores	Convergencia lenta en redes grandes
Admite le balance de cargas	Solo admite balance de cargas de coste equivalente, el cuello de botella suele ser un inconveniente
Generalmente está libre de bucles	Generalmente está libre de bucles
	Escalabilidad limitada (15 hops)
	No tiene en cuenta el ancho de banda, el retardo ni la fiabilidad a la hora de aplicar la métrica
	No soporta VLSM (en su versión 1) Mascara de subred de longitud variable
	Las actualizaciones difundidas pueden provocar un derroche masivo de ciclos de UPC en los servidores
	NO admite actualizaciones autenticadas, lo que significa que un enrutador molesto podría perturbar el funcionamiento de las rutas

redes grandes y complejas con buenos principios de diseño, mientras que RIP está destinado a redes pequeñas en las cuales un único protocolo puede reducir el tiempo de configuración y diseño[7].Si la red es lo suficientemente pequeña para utilizar el protocolo OSPF. En su lugar, es más probable tener que seguir utilizando RIP o quizás cambiara EIGRP, o a un sistema de enrutamiento estático. Sin embargo, a pesar de estas diferencias, se ha establecido una comparación de ambos sistemas; para no volver a mencionar de nuevo las características de RIP, se mencionan las ventajas de OSPF sobre RIP, así:

- OSPF es mucho más escalable que RIP
- Admite VLSM (contrario a lo que ocurre en RIPv1)
- Presenta menor utilización de red para lograr redes bastante estables
- Dispone de una mejor selección de camino
- Evita de forma elegante los bucles de enrutamiento
- Utiliza una métrica mucho más útil
- Dispone de un diseño jerárquico (no funciona muy bien con estructura IP pobremente diseñada)
- La convergencia es más rápida

Una desventaja de OSPF sobre RIP es el requerimiento de más potencia de procesador y memoria; además, requiere más tiempo de diseño e implementación.

10. Análisis de resultados

Conocidas las características más relevantes de RIP, se considera un escenario en donde se encontrará una LAN a cada extremo con un único camino demás de15 enrutadores. Entonces ¿Sería posible que en una primera emisión de transmisión los enrutadores establecieran una convergencia en esta red? Así se supone que no se puede alcanzar convergencia en esta red de extremo a extremo, debido que la métrica de un destino se calcula como la métrica comunicada por un vecino, más la distancia en alcanzar a ese vecino. Teniendo en cuenta el límite de 15 saltos mencionado anteriormente y el enrutador solo puede enrutar si el camino se presenta en su tabla de enrutamiento.

A continuación se presenta el escenario (Figura 16) y su configuración inicial, establecida sobre BOSONNetSim v5.27[7]:

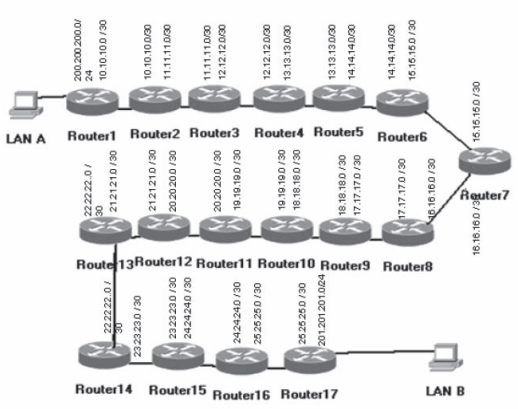

Figura 16. Escenario y configuración de enrutadores.

[7] Se puede descargar una versión BETA del software en http://www.boson.com/netsim.

```
ROUTER1
Gateway of last resort is not set
C200.200.200.0/24 is directly connected, Ethernet0
C10.10.10.0/30 is directly connected, Serial0
R11.11.11.0/30 [120/1] via 10.10.10.2, 00:02:24, Serial0
R12.12.12.0/30 [120/2] via 10.10.10.2, 00:07:22, Serial0
R13.13.13.0/30 [120/3] via 10.10.10.2, 00:07:43, Serial0
R14.14.14.0/30 [120/4] via 10.10.10.2, 00:03:19, Serial0
R15.15.15.0/30 [120/5] via 10.10.10.2, 00:05:14, Serial0
R16.16.16.0/30 [120/6] via 10.10.10.2, 00:04:31, Serial0
R17.17.17.0/30 [120/7] via 10.10.10.2, 00:06:35, Serial0
  R18.18.18.0/30 [120/8] via 10.10.10.2, 00:08:14, Serial0
R19.19.19.0/30 [120/9] via 10.10.10.2, 00:01:27, Serial0
R20.20.20.0/30 [120/10] via 10.10.10.2, 00:03:33, Serial0
R21.21.21.0/30 [120/11] via 10.10.10.2, 00:08:41, Serial0
R22.22.22.0/30 [120/12] via 10.10.10.2, 00:03:23, Serial0
R23.23.23.0/30 [120/13] via 10.10.10.2, 00:09:39, Serial0
R24.24.24.0/30 [120/14] via 10.10.10.2, 00:02:16, Serial0
```

Figura 17. Datos del enrutador 1.

Utilizando el comando *show iproute*, se observa en la anterior tabla que en el *router*1 (Figura 17)no se representa el camino a LAN B (ubicada sobre la red 201.201.201.0/24) y por ello no es posible alcanzar comunicación inmediata. Si hacemos un seguimiento por cada enrutador observamos que hasta el enrutador 4 podemos alcanzar esta red, como lo vemos en la siguiente tabla que nos dice que vía 13.13.13.2, que es la conexión serial1, conectada de manera directa al enrutador 4.

```
ROUTER4
Gateway of last resort is not set
C12.12.12.0/30 is directly connected, Serial0
R11.11.11.0/30 [120/1] via 12.12.12.1, 00:09:24, Serial0
R10.10.10.0/30 [120/2] via 12.12.12.1, 00:04:30, Serial0
R200.200.200.0/24 [120/3] via 12.12.12.1, 00:01:41, Serial0
C13.13.13.0/30 is directly connected, Serial1
R14.14.14.0/30 [120/1] via 13.13.13.2, 00:08:13, Serial1
R15.15.15.0/30 [120/2] via 13.13.13.2, 00:01:32, Serial1
R16.16.16.0/30 [120/3] via 13.13.13.2, 00:08:26, Serial1
R17.17.17.0/30 [120/4] via 13.13.13.2, 00:08:40, Serial1
R18.18.18.0/30 [120/5] via 13.13.13.2, 00:06:16, Serial1
R19.19.19.0/30 [120/6] via 13.13.13.2, 00:06:25, Serial1
R20.20.20.0/30 [120/7] via 13.13.13.2, 00:05:28, Serial1
R21.21.21.0/30 [120/8] via 13.13.13.2, 00:07:32, Serial1
R22.22.22.0/30 [120/9] via 13.13.13.2, 00:08:44, Serial1
R23.23.23.0/30 [120/10] via 13.13.13.2, 00:02:23, Serial1
R24.24.24.0/30 [120/11] via 13.13.13.2, 00:04:30, Serial1
R25.25.25.0/30 [120/12] via 13.13.13.2, 00:07:38, Serial1
R 201.201.201.0/24 [120/13] via 13.13.13.2, 00:02:39, Serial1
```

Figura 18. Datos del enrutador 4.

Se observa que sí es posible una comunicación con el otro extremo desde el enrutador 4 (Figura 17). Otra hipótesis: si el *router*1 alcanza el *router*4 entonces el *router*1 alcanza hasta el *router*17.

```
ROUTER1
Gateway of last resort is not set
C200.200.200.0/24 is directly connected, Ethernet0
C10.10.10.0/30 is directly connected, Serial0
R11.11.11.0/30 [120/1] via 10.10.10.2, 00:06:28, Serial0
R12.12.12.0/30 [120/2] via 10.10.10.2, 00:07:22, Serial0
R13.13.13.0/30 [120/3] via 10.10.10.2, 00:01:30, Serial0
R14.14.14.0/30 [120/4] via 10.10.10.2, 00:05:32, Serial0
R15.15.15.0/30 [120/5] via 10.10.10.2, 00:02:15, Serial0
R16.16.16.0/30 [120/6] via 10.10.10.2, 00:04:28, Serial0
R17.17.17.0/30 [120/7] via 10.10.10.2, 00:03:22, Serial0
R18.18.18.0/30 [120/8] via 10.10.10.2, 00:08:40, Serial0
R19.19.19.0/30 [120/9] via 10.10.10.2, 00:05:13, Serial0
R20.20.20.0/30 [120/10] via 10.10.10.2, 00:03:13, Serial0
R21.21.21.0/30 [120/11] via 10.10.10.2, 00:03:27, Serial0
R22.22.22.0/30 [120/12] via 10.10.10.2, 00:01:36, Serial0
R23.23.23.0/30 [120/13] via 10.10.10.2, 00:01:12, Serial0
R24.24.24.0/30 [120/14] via 10.10.10.2, 00:02:15, Serial0
R25.25.25.0/30 [120/15] via 10.10.10.2, 00:02:33, Serial0
R 201.201.201.0/24 [120/16] via 10.10.10.2, 00:03:31, Serial0
```

Figura 19. Caminos agregados en el enrutador 1.

Al realizar de nuevo el seguimiento, vemos que el enrutador 1 (Figura 19) ya presenta un camino a 201.201.201.0/24. Este suceso se debe a un nuevo cálculo en el alcance de rutas; es decir, cuando un mensaje de LAN_A a LAN_B llega al *router*4, este reconoce el destino y actualiza esta ruta para todos los *routers* que pasaron desde el *router*1.Así, en este escenario, todos los *routers* son alcanzables, superando la "limitación" de 15 saltos.

11. Conclusiones

RIP es usado como un protocolo de enrutamiento de pasarela interior, es decir que se utiliza en escenarios "pequeños" debido a las limitaciones que presenta sobre escenarios complejos; varias de las falencias han sido solucionadas con la segunda versión de este protocolo. Estas se pueden implementar en consideraciones deIPv6, que es protocolo descrito en RIPngfor IPv6, del *request for*

comments 2080 [7], por lo cual la implementación de RIP es la primera alternativa en los entornos de redes por la gran flexibilidad que presenta en su sencillez y eficacia.

El conocimiento de este tipo de enrutamiento hace que el administrador de red obtenga la capacidad para determinar qué opciones configura sobre una red particular. Es decir, conoce los costos de configuración sobre la red del protocolo y puede determinar la mejor solución de inconvenientes que pueda presentar el algoritmo en el cual se basa este protocolo y cómo puede mezclar un enrutamiento basado en vector de distancia o de estado de enlace[8]. En el soporte que ofrece Cisco se encuentra una documentación adecuada que es analizada bajo una serie de escenarios comunes.

Es interesante observar la evolución de RIP que empieza en 1970 y se estandariza en 1988 (rfc 1058), presentando extensiones que lo mejoran en 1994 (rfc 1723) y una adaptación a IPv6 en 1997(rfc 2080); pese a estas modificaciones, no cambia su estructura básica.

Referencias bibliográficas

[1] B. Hill, "Manual de referencia CISCO. McGraw-Hill, pp. 631-700, 2002.

[2] *Routing Information Protocol*, RFC1058. Jun 1988

[3] Internet official Protocol Standards, RFC1720. Junio 1994.

[4] G. Malkin, "RIP v2,Carring Additional" RFC1723 "Request for Comments: 1723", pp. 0-7, Nov. 1994.

[5] CONFIGURING RIP, [En línea]. Disponible: www.cisco.com.

[6] RIP para IP, [En línea]. Disponible: http://www.microsoft.com/spain.

[7] G. Malkin.R. Minnear. "RIPng for IPv6 " RFC2080.

[8] RIPng for IPv6, RFC 2080. Enero 1997.

[9] G. Meyer. S. Sherry "Triggered Extensions to RIP to Support Demand Circuits" RFC 2091 "Request for Comments: 2091", pp. 1-21, Ene. 1997.

[10] G. Malkin. "RIP Version 2 Protocol analysis" RFC 1387 "Request for Comments: 1387". pp. 1-3, Ene. 1997.

[8] Este algoritmo determina la lógica de enrutamiento de estado de enlace que funciona en el algoritmo primero, la ruta más corta (SPFShortestPathFirst) de Dykstra, usado por OSPF (Open ShortestPathFirst). Este tema no compete de manera directa a este documento pero se convierte en un requerimiento para la comunicación de AS (sistemas autónomos) híbridos de enrutamiento.